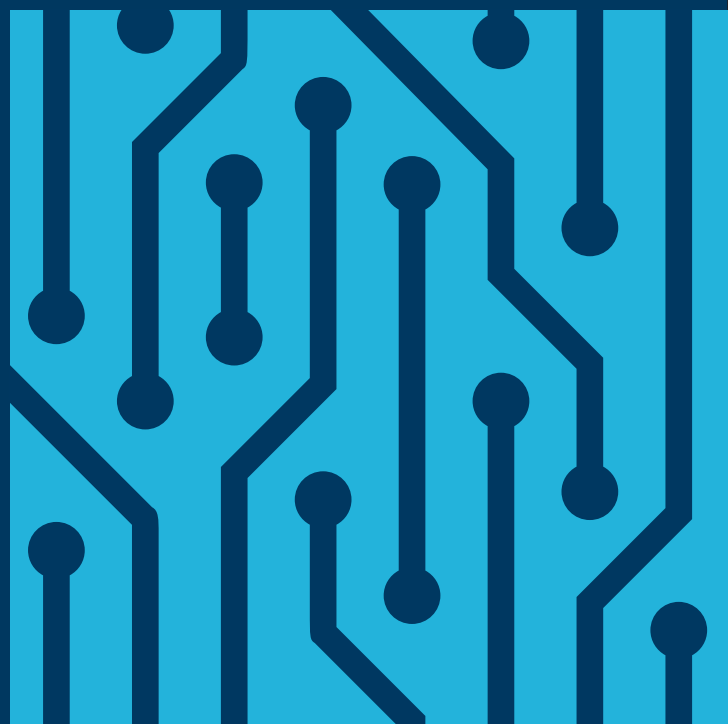


Application Note #23

Binary Measurement Data Transfer



Safety Instructions



Before starting operation, read the user manual, in particular the General Safety Instructions and the safety instructions for connecting the test object!

In the development of automated measurement systems using electronic loads, fast and efficient data transfer between the load and the controlling PC is crucial. Binary block data transfer according to IEEE 488.2 offers significantly higher speed, a more compact data format, and more efficient processing of floating-point values compared to ASCII transmission. This application note describes the implementation and configuration of binary measurement data transfer with H&H electronic loads across different data interfaces, including relevant requirements.

Principles of Binary Block Transfer

The binary data transfer is supported by USB, LAN, and RS-232 interfaces. GPIB and CAN cannot be used for this purpose. Block data transfer is exclusively applied when using the query command DATA:REM?.

The IEEE 488.2 standard defines a compact data format for binary data blocks (see section 8.7.9: Definite Length Arbitrary Block Response Data).

Structure:

1	2	3	4	5
#	Number of elements that follow (ASCII) <nonzero digit>	Number of bytes that follow (ASCII) <digit(s)>	Binary data (non-ASCII) <8 bit data bytes>	Termination character

Field description of the data structure:

#<n><nnn...><data><NL>

#	Start Character
<n>	Number of subsequent ASCII digits specifying the data block length
<nnn...>	Number of data bytes (in ASCII digits)
<data>	Measurement Data (IEEE 754 floating-point format)
<NL>	Termination Character (NL, 0xA)



Example:

#3144<data>NL

3: three digits are used to specify the data length

144: 144 data bytes follow

<data>: 36 floating-point values in IEEE 754 format (REAL,32)

Tip:

On the PC, the received binary block data requires further processing.

Software solutions such as pyVISA provide native support for receiving and interpreting binary block data, enabling seamless integration into automated measurement systems.

Enabling Binary Block Transfer

To activate binary data mode, the following SCPI commands are relevant:

FORMat:DATA REAL,32	enables 32-bit floating point format (IEEE 754)
FORMat:BORDER NORM	Little Endian
FORMat:BORDER SWAP	Big Endian

Depending on the required byte order (endianness), either NORM or SWAP must be used.

These settings are volatile and will be reset after a power cycle or device reset.

Data Query with DATA:REMove?

Measurement data is queried using the SCPI command: DATA:REMove?

Where <NRf> specifies the number of data records to query (1...100).

The earliest records are transferred first (first in, first out).

Each measurement point consists of: <timestamp>, <measured voltage>, <measured current>.

The data read is immediately deleted in the electronic load.

Example:

DATA:POIN?	Query total number of stored measurement records 307 is returned
DATA:REM? 100	read the first 100 records
DATA:REM? 100	read the next 100 records
DATA:REM? 100	read the next 100 records
DATA:REM? 7	read the last 7 records

